

DOCUMENT RESUME

ED 250 175

SE 045 151

AUTHOR Linn, Marcia C.
TITLE Policy Issues in Computer Education. Assessing the Cognitive Consequences of Computer Environments for Learning (ACCCEL).
INSTITUTION California Univ., Berkeley. Lawrence Hall of Science.
SPONS AGENCY National Inst. of Education (ED), Washington, DC.
PUB DATE 84
GRANT 400-83-0017
NOTE 44p.; Paper presented at the Annual Meeting of the American Educational Research Association (68th, New Orleans, LA, April, 1984).
PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)
EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Cognitive Processes; *Computer Assisted Instruction; *Computer Science Education; *Computer Software; Educational Research; Instructional Innovation; Instructional Material Evaluation; Junior High Schools; *Policy Formation; Problem Solving; *Skill Development; Student Motivation
IDENTIFIERS *Computer Uses in Education

ABSTRACT

This paper analyzes the capabilities of the computer learning environment identified by the Assessing the Cognitive Consequences of Computer Environments for Learning (ACCCEL) Project, augments the analysis with experimental work, and discusses how schools can implement policies which provide for the maximum potential of computers. The ACCCEL project spent 2 years in over 25 junior high school sites and: (1) identified and explored features of the computer environment which enhance higher cognitive skills; (2) gained a clear sense of student deficiencies in the skills needed to master formal systems (the cyclical procedural skills of planning, testing, and debugging); (3) developed a model of autonomous learning which follows this procedural skills cycle; (4) performed an in-depth analysis of student interaction with the problem-solving software game "Rocky's Boots"; (5) worked closely with classroom teachers to discover their needs and concerns regarding curricula which integrate software; (6) developed software and curricula to emphasize higher cognitive skills during programming instruction; and (7) characterized which students profit from different computer learning environments. Policy implications discussed focus on fostering basic and higher cognitive skills, providing dynamic support for teachers, maximizing teacher effectiveness, and motivating students. (JN)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

ED250175

Policy Issues in Computer Education

Marcia C. Linn

Assessing the Cognitive Consequences of Computer Environments in Learning (ACCCEL)

Lawrence Hall of Science
University of California
Berkeley, CA 94720

This material is based upon research supported by the National Institute of Education under Grant No. 400-83-0017. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Institute of Education.

This paper was presented at the annual meeting of the American Educational Research Association, New Orleans, 1984

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

✓ This document has been reproduced as received from the person or organization originating it.

□ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

Marcia C. Linn

BEST COPY AVAILABLE

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)

SE045151

SUMMARY OF ACCCEL PROJECT

The Assessing the Cognitive Consequences of Computer Environments for Learning (ACCCEL) Project funded by the National Institute of Education jointly conducted by the Lawrence Hall of Science at the University of California at Berkeley and the Far West Laboratories in San Francisco. The primary objective is to examine the potential of computer environments for fostering cognitive understanding through classroom observations and empirical studies.

The ACCCEL project has identified six features of the computer learning environment which make it particularly cognitively demanding. First, the environment is interactive. The computer can respond quickly and informatively to the learner. Thus, individuals can try several possible repairs for errors in their programs and obtain immediate responses from the computer. In contrast, it often takes days or weeks to get responses to homework. Second, the computer can provide precise feedback, e.g. information about where the problem solver has gone wrong. Third, this preciseness is reproducible, providing consistent responses to all learners. Thus, learners are treated equally, receiving the same response from the computer every time they give the same input. Fourth, the environment for learning computer skills is complex. The computer environment can strain the processing capacity of the learner. Fifth, computers can provide learners with problems which have multiple solutions. Thus, students can write several programs to do the same task and compare the effectiveness of their programs. Sixth, the computer environment is challenging. Computer learning environments can motivate the learner to solve complex problems which might otherwise be omitted from the curriculum.

The Project has spent two years in over 25 junior high sites and has: (1) identified and explored features of the computer environment which enhance higher cognitive skills; (2) gained a clear sense of student deficiencies in the skills necessary to master formal systems (the cyclical procedural skills of planning, testing, and debugging); (3) developed a model of autonomous learning which follows this procedural skills cycle; (4) performed an in-depth analysis of student interaction with the problem-solving software game "Kocky's Boots"; (5) worked closely with classroom teachers to discover their needs and concerns regarding curricula which integrate software; (6) developed software and curricula to emphasize higher cognitive skills during programming instruction; and (7) characterized which students profit from different computer learning environments.

ACCCEL Research Reports

Assessing the Cognitive Consequences of Computer Environments for Learning (ACCCEL)

Marcia C. Linn - Principal Investigator
Lawrence Hall of Science

Charles Fisher - Co-Director
Far West Laboratory for
Educational Research and Development

NIE Contract DE 0400 B3 0017

1. Linn, M.C., Fisher, C., Mandinach, E.B., Dalbey, J., & Beckum, L. ACCCEL Proposal - Demands and cognitive consequences of computer learning - 1982. 155 pgs. Fee: \$15.00
2. Linn, M.C. & Fisher, C.W. Computer education: The gap between promise and reality, Proceedings of Making Our Schools More Effective: A Conference for Educators, Far West Laboratory, San Francisco, 1982. 20 pgs. Fee: \$2.00
3. Dalbey, J. & Linn, M.C. The demands and requirements of computer programming, A review of the literature, 1983. 32 pgs. Fee: \$3.00
4. Mandinach, E.B. & Fisher, C.W. Review of research on the cognitive effects of computer-assisted learning, 1983. 82 pgs. Fee: \$7.00
5. Linn, M.C. Fostering equitable consequences from computer learning environments. Paper presented at the AERA SIG: Research on Women in Education, Mid-Year conference, Tempe, 1983. ~~See Boles~~, 1984. 32 pgs. Fee: \$3.00 6. Dalbey, J. Spider World Reference Manual and Teacher's Guide, 1983. 58 pgs. and disk. Fee: \$15.00
7. Dalbey, J. Teaching programming with Spider World. Paper presented at the seventh Annual Western Educational Computing Conference, San Francisco, November 18, 1983. 7 pgs. FREE

8. Linn, M.C., Fisher, C.W., Dalbey, J., Mandinach, E.B., Burbules, N.C., & Reese, P. The effects of learning to program and learning from cognitively demanding software. Research design, 1983. 60 pgs. Fee: \$6.00
9. Mandinach, E.B. Sex differences in a computer game of logical reasoning: The WUMPUS study. Paper presented at the AERA SIG: Research on Women in Education, Mid-Year Conference, Tempe, 1983. 13 pgs. FREE
10. Linn, M.C. & Stein, J. The gap between promise and reality in classroom computing instruction, 1984. 11 pgs. FREE
11. Burbules, N. & Reese, P. Teaching logic to children: An exploratory study of "Rocky's boots," 1984. 40 pgs. Fee: \$4.00
12. Mandinach, E.B. The role of strategic knowledge and self-regulation in learning to solve an intellectual computer game. Unpublished doctoral dissertation, Stanford University, July 1984.
13. Mandinach, E.B. Selection of software for studies of acquisition of higher cognitive skills from computer assisted learning, 1984. 30 pgs. Fee: \$3.00
14. Mandinach, E.B. The role of self-regulation in CAI. Presented at AERA, 1984. 33 pgs. Fee: \$3.00
15. Mandinach, E.B. Clarifying the "A" in CAI for learners of different abilities. Presented at AERA/MARST, 1984. 33 pgs. Fee: \$3.00
16. Mandinach, E.B., & Corno, L. Cognitive engagement and individual differences in a computer problem solving task. Sex Roles, 1984. 32 pgs. Fee: \$3.00
17. Stein, J. & Reese, P. Rocky's Boots Curriculum, 1984. (In preparation)
18. Linn, M.C. Fostering equitable consequences from computer learning environments: An overview, 1984. 35 pgs. Fee: \$3.00
19. Dalbey, J., Tourniaire, F., Linn, M.C. Making programming instruction cognitively demanding: An intervention study. Paper presented at AERA, 1984. Fee: \$2.00
20. Dalbey, J., Linn, M.C. Spider World: A robot language for learning to program. Paper presented at AERA, 1984. Fee: \$2.00
21. Linn, M.C. Policy Issues in Computer Education. Paper presented in AERA, 1984. Fee: \$3.00

* Fees cover duplication and mailing expenses only.
Please make checks payable to Regents of UC.

October, 1984

INTRODUCTION

Computers play an increasingly central role in our society. Many college presidents are promising a computer in every student room. A recent Gallup poll revealed that 72% of the population feels we should require high school computer courses (Classroom Computer Learning, 1984). Microcomputers are now available in a large number of our nation's schools and most students in public education in the United States will have the opportunity to interact with the computer before graduating from high school (Becker, 1984). It is almost universally agreed that computer learning environments deserve a role in public education yet determining that role requires considerable analysis and experimentation. Whether the computer will serve as a tool, a new medium of communication, or a savior of education remains to be determined.

This paper analyzes the capabilities of computer learning environments, augments the analysis with experimental work and discusses how schools can implement policies which provide for the maximum potential of computers. The areas considered are: a) fostering higher cognitive skills, b) fostering basic skills, c) providing dynamic support for learners, d) maximizing teacher effectiveness, and e) motivating learners.

Perceived Roles for Computers in Education

Attempts to assess the possible advantages of the computer learning environment often offer more rhetoric than reason. Most responses can be characterized by one of three viewpoints concerning microcomputers in education: the computer as a tool, the computer as an innovative communication medium, and the computer as a savior.

Those who see the computer as a tool emphasize the ability of computers to perform well-established instructional tasks more efficiently or for less cost. They do not expect computers to lead to qualitative changes in the educational process. Just as the pencil increased the amount of rewriting and the ditto machine increased the number of student worksheets, so the computer can be seen as an efficient way to administer problems, to correct papers, and to keep track of student progress. As a tool, the computer has been most widely used for drill and practice on basic skills such as arithmetic computation. Other tool uses include wordprocessing and data-base management.

Does the computer, in addition, offer a new medium for instruction? Can the computer lead to qualitative changes in classroom learning? Analysis of the capabilities of computers combined with the development of products to take advantage of these capabilities is required to answer this question.

Such an analysis is difficult for at least two reasons. First, creative and effective uses for new technologies are rarely apparent when they are first developed. For example, when motion pictures first became available, plays were filmed from a single camera position. In contrast, today, film producers know thousands of techniques for creating special effects which often astound audiences. Similarly, educational television was often used to film lectures at the onset and is now often used to show details of intricate surgery, to create visual images of difficult concepts, etc.

Second, even the features of a new learning environment which make it special and unique may, at first, go unnoticed. The unintended effects of new programs occasionally overshadow their intended consequences. For example, professional development programs designed to help teachers use new curricula are often found to have at least as much impact on classroom organization and innovation in areas not covered by the course. Analysis reveals that this outcome results from opportunities for teachers to get together, watch each other teach, and share ideas. Similarly, a toy library program for mothers of preschool age children that was intended to enrich the world of children by providing toys, was at least as effective in enriching the lives of mothers by giving them access to other mothers with similar concerns.

For some, the computer is seen as a savior of the edu-

cational enterprise (e.g. Carbonell, 1970; Papert, 1981; Taylor, 1980). These individuals proclaim that computers will revolutionize human learning and dramatically improve the quality of classroom instruction. This is not a new view. The tendency to view the computer as a savior for education began with the introduction of main frame and mini computers in the fifties and sixties and is being repeated with the introduction of the microcomputers of the seventies and eighties. For example, Papert in *Mindstorms* (1981) promised that computers would literally change the thinking patterns of students. Papert's prose often motivates teachers to believe that major changes in problem solving skills are just within students' grasp. In contrast, many point out that placing computers in classrooms is hardly enough to lead to intellectual breakthroughs. After all, placing pianos in homes is not sufficient to get students to learn how to play. Other forecasters promise that the computer can replace or largely supplant the activities of teachers, by using natural language communication capabilities or artificial intelligence techniques to build models of the learner (e.g. Aaronson, 1984; Carbonell, 1970). In contrast, observers note that the ability of computers to process natural language and behave intelligently is progressing relatively slowly and may never impact on education. Commentators have warned of the danger of "techno-romanticism" which may result in disappointment among those who believe these possibly overstated claims (Pea, 1984).

Thus, no universally agreed-upon role for computers in education as yet exists. This paper provides an analysis of the capabilities of computers combined with examination of some innovative uses to clarify appropriate roles is appropriate and suggest policies for schools which capitalize on this knowledge.

ANALYZING THE COMPUTER LEARNING ENVIRONMENT

The Assessing the Cognitive Consequences of Computer Environments for Learning (ACCCEL) project jointly carried out by the Lawrence Hall of Science at the University of California at Berkeley and the Far West Laboratory in San Francisco, has analyzed the potential of the computer learning environment and assessed the implications of that analysis in a series of empirical studies (e.g. Dalbey, Tournaire, & Linn, 1984; Linn & Fisher, 1983; Linn & Stein, in press; Mandinach, 1984). In addition, since computers have been used by specialists for some time but are only now within the budgets of schools, we can also gain insights into how to use this new technology in education by examining its use by experts. These analyses clarify the potential role of the computer in education and suggest appropriate policies for schools seeking to maximize this potential.

Features of the Computer Learning Environment

The ACCCEL project has identified six features of the

computer learning environment which make it particularly well suited to educational innovation (Linn & Fisher, 1984). First, the computer learning environment is interactive. The computer can respond to the learner quickly and informatively. Thus, individuals can interrogate a learning environment and obtain immediate responses. In contrast, students often wait days or weeks to get feedback on their homework assignments. Second, the computer can provide precise feedback or information about the users' input. In contrast, students in classrooms often get a letter grade like "B-", or perhaps unreliable feedback when papers are graded by other students, rather than getting precise information about their efforts. Third, the computer's preciseness is reproducible. The computer provides the same response to all learners and the same response every time the learner behaves in a particular fashion. Thus learners are treated equally and learners who question the accuracy of feedback or forget the computer's response can recreate it easily. Fourth, the computer learning environment is complex. The computer can assess students' response to problems which strain their processing capacity. Thus, the computer can analyze detailed computer programs and provide precise information about the flow of information through the program and about violations of the rules for the computer language. Fifth, computers can assess responses to problems which have multiple solutions. In fact, in computer learning environments, students can compare different solu-

tions to the same problem using a variety of criteria including efficiency and generalize ability. Sixth, the computer environment is challenging. The computer can motivate learners to solve complex problems by providing hints and feedback which keep the learner engaged in the task.

Capitalizing on the Computer Learning Environment

We are only beginning to discover ways to capitalize on the features of the computer learning environment to design educational programs which tap the potential of computers in education (e.g. Linn, Fisher, Beckun, Dalbey, Mandinach, 1982; Nickerson, 1984). Some have examined the potential for teaching basic skills such as reading (e.g. Frederickson et. al., 1984), while others have sought opportunities for teaching higher cognitive skills often referred to as problem solving skills (e.g. Anderson, Farrell, & Sauers, in press; Pea & Kurland, 1983; Linn et. al., 1983). Still others have sought to clarify the role of the computer in providing dynamic support for the learner and in increasing the teachers' effectiveness (e.g. Sleeman & Brown, 1982).

These investigations are complex because it is unclear exactly what to compare the computer to. This issue has permeated educational research on all new technologies (e.g. Salomon & Gardner, 1984). For example, even when used effectively, new technologies are unlikely to teach "better" than a skilled tutor (e.g. Anderson, et. al., in press).

Furthermore, with enough time and energy, any instructional procedure presented by computer can probably be presented without the computer. Thus, we are concerned not with whether the computer can teach the unteachable but with how the computer communicates the complex, and presents the perplexing. Computer advantages must be assessed in terms of tradeoffs. What are teachers freed to do when computers are used? Can the motivational features of the computer attract otherwise bored students? Effective uses of computers for education have not yet been fully realized.

Fostering Higher Cognitive Skills

Computer learning environments can be used to increase direct instruction in the area of higher cognitive skills. Many highly publicized reports have called attention to the deplorable state of science and mathematics education in the United States today and emphasized the need for instruction in higher cognitive skills. The report of the National Commission on Excellence in Education, "A Nation at Risk", (1983) emphasizes the pressing need for educational reform to create a "learning society," a society which can solve new naturally occurring problems. Similarly, "Educating Americans for the Twenty-First Century", (National Science Board, 1984) has called for the "new basics" or the thinking skills students need to cope with rapid technological and scientific advances. Both reports emphasize the need for instruction which fosters problem solving, prepares learners

to deal with problems arising as a result of technological change, and encourages students to think critically.

The ACCCEL project has illustrated how the computer learning environment can increase the quantity and quality of instruction in these areas. School policies which result in such increases and make additional increases likely are desirable.

The ACCCEL project has identified two higher cognitive skills which are particularly likely to be fostered by the features of the computer learning environment. The first is planning. Planning refers to the process of taking components and combining them to come up with a unique solution to a problem. Computer learning environments typically involve a set of primitives or components for solving problems. Students reflect on how to combine these primitives to come up with a problem solution. In so doing, they engage in planning. Once they come up with a plan, they can implement it on the computer and get feedback on its success.

For example, students wishing to understand the relationship between the equation for a line and its slope can come up with a plan of plotting the equations for six different lines and implement their plan using a piece of educational software called *Algebra Arcade* (1983; Dugdale, 1982) (see Appendix A). They get immediate feedback for their efforts. In another example, students can devise a

computer program as a plan for solving a problem such as "alphabetize these words". When the program is written, immediate feedback on the effectiveness of the plan is available. These examples illustrate the potential role of the computer in education. In a typical class without a computer, students are unlikely to get immediate feedback on their plans and when they do get feedback it is apt to be less precise than the computer feedback. Furthermore, the computer environment provides opportunities to attempt problems with multiple solutions and get feedback on the approach chosen. Thus, the computer may well qualitatively change instruction in the higher cognitive skill of planning.

A second higher cognitive skill which can be fostered by the computer learning environment is testing. Testing refers to the ability to determine whether one's plan accomplishes the planned objectives. Many times, the initial trial of a computer plan does not execute as intended and it is clear that a problem exists. At other times, the plan executes and is not clear whether the plan is accurate. In both cases, students need skill in testing their plans to determine the conditions under which they work. For example, students may need to test the boundary conditions for their computer programs. Skill in testing a plan is facilitated in the computer learning environment because of the precise feedback available. Thus, students can design tests to capitalize on the feedback features of the learning

environment.

Examination of the problem solving skills of experts using the computer provides further examples of skills likely to be fostered by the computer learning environment. Several studies of expert programmers reveal these skills (e.g. Kurland & Pea, 1984; Jeffries, Turner, Polson, & Atwood, 1981; McKiethan, Reitman, Rueter, & Hirtle, 1981).

One component of programming expertise is the acquisition of an extensive repertoire of "programming templates" (see Table 1). Templates are stereotypic prescriptions for a particular aspect of a program, similar to schemata as described by Norman & Rumelhardt (1975). They include both algorithms and heuristics. Templates can apply to a whole program as exemplified in a "input-process-output" template. Templates can also apply to a specific function of a program such as a loop function. Research by Pea & Kurland (1983) reveals that both expert adult and expert student programmers can articulate their templates, recognize the relationships between their templates and new templates, and actively seek new templates.

The computer learning environment can illustrate the value of templates as well as providing opportunities for students to develop powerful templates. Most commonly, templates are acquired by using programs written by others. These programs are reformulated to solve new problems by the learner. Thus, students can acquire and refine powerful

templates and also develop some awareness of the role of templates in problem solving through interaction with computer learning environments.

For example, students rarely invent procedures for doing computer sorting but instead take sort routines written by others and alter them for their needs. The process of altering sort routines allows students to elaborate their templates, developing a more robust understanding. The process also illustrates an important programming technique regularly used by experts. Experts generally reformulate existing code to suit new situations, or engage in what is called "code bunning".

At present, higher cognitive skills are most commonly encountered in programming rather than when using the computer to solve problems from other subjects. Much recent research in science education, however, reveals that higher cognitive skills are often developed in specific subject matter domains and are difficult to generalize to new domains (Linn, in press; Linn, 1984). It seems likely that students need to exercise higher cognitive skills in a wide range of subject matter areas before they are likely to transfer them to new areas. In addition, we need instruction that develops these higher cognitive skills in subject matter areas of particular importance to citizens in our society. To increase emphasis in the curriculum on higher cognitive skills, we must tap the potential of the computer

learning environment.

Enhancing Basic Skills

Computer learning environments provide opportunities for fostering basic skills as well as higher cognitive skills. Software developers are now making substantial progress in determining how to use the precise and interactive features of the environment effectively (e.g. Chomsky, 1984; Keil, 1983). For example, Frederickson et. al. (1984) have task-analyzed basic reading behavior of slow readers and have developed programs to provide practice in needed skills such as recognizing whole words from partial cues, using context to determine a missing word, and reading phrases instead of individual words. This software takes advantage of the graphics capability of the computer to provide practice on skills not easily practiced in other ways.

Providing Dynamic Support for Learners

A third area where the features of the computer learning environment might be capitalized upon is in providing dynamic support for learners. Dynamic support can be provided by tailoring the learning environment to the needs of the student. Computer learning environments are particularly well suited to such tailoring because they provide multiple paths to the same outcome, because they can diagnose some problems and save students from unproductive perusal of wrong ideas, because they can constrain behavior

ad delineate alternatives, and because they can provide practice in selecting an appropriate course of action. Thus, learners who require extra practice on a particular component of a formal system can receive that practice while other students move on to other activities. Furthermore, students who learn better under one set of conditions than another can select the conditions which appeal to them or can be guided by either the computer or the teacher to identify and use information tailored to their needs (e.g. di Sessa, 1984; Mandinach, 1984; Reil, Levin & Miller-Souviney, 1984).

For example, many pieces of educational software provide multiple modes for interaction. The Spider World software developed by a member of the ACCCEL staff (Dalbey, 1984) provides two different problem solving modes, draw mode and list mode (see Appendix A for more details). In draw mode, students can construct the solution to a problem incrementally and visually. Once they have constructed the solution, they can examine the instructions used and look for patterns which they might combine into subroutines to increase the efficiency of their solution. Alternatively, students can solve problems in list mode by writing a computer program to instruct the computer to construct the figure they wish to create. The opportunity of using multiple modes takes advantage of the computer learning environment by adapting to the needs of the learner.

Another way the dynamic support for the learner can be advantageous in educational settings is in fostering student autonomy. Students encouraged to take advantage of learning environments but not specifically required to follow a single approach can develop skill in determining which approach is best for them. Thus, in the Spider World example above, students might use the draw mode option initially when they have no clues as to how to proceed in solving a problem but might decide as time went on to solve problems using list mode and avoid reliance on draw mode. Using list mode is more efficient once one has mastered the skills required. In making these decisions on their own, students gain feedback about their reasoning processes and are likely to become more effective autonomous learners. Thus, the computer learning environment has the potential for fostering autonomous learning by offering students choices in the types of feedback and avenues for solutions to problems.

Another example of how the computer provides dynamic support comes from observing the behavior of expert programmers. Experts capitalize on the precision of the computer environment and interact frequently with its precise feedback in order to test the system, understand how it works, and make it suit their needs. In essence, experts are responsible about their own learning, use feedback from the learning situation to guide subsequent activities, select sub-goals, and ultimately come up with the solution to a problem. This responsibility which leads to autonomous

learning, has been discussed extensively by those concerned with instruction (Glaser, 1984; Corno & Mandinach, 1983; Thomas and Rohwer, 1982). Computer learning environments are ideally suited to the needs of autonomous learners.

Maximizing Teacher Effectiveness

A fourth way that the computer learning environment can be capitalized upon is by maximizing teacher effectiveness. The computer can provide many types of instruction as well as the dynamic support for the learner described above. This frees the teacher to devote energy to fostering the skills required to use the environment effectively.

Just as students need instruction in how to use reference materials and library services efficiently, so they need considerable help in learning how to use the computer learning environment resourcefully. Often, students are so focused on solving the problem at hand that they have difficulty learning procedures for using the richness of the computer environment. Students often ask the teacher to tell them the answer when they have problems using computers. However, if instead of supplying answers the teacher helps the students develop planning and testing procedures, both students and teachers will benefit. Students will learn higher cognitive skills and teachers will serve as facilitators rather than encyclopedias.

For example, one teacher observed by the ACCCEL project

invoked the rule that students needed to spend ten minutes using testing to attempt to solve a problem, when learning to program the computer, before calling on the teacher. As a result, students developed alternative hypotheses for solving problems and then tested their hypotheses. In addition, to make this requirement effective, when the teacher assisted students, she typically provided them with information about how they could improve their testing skills to get a better idea of what was wrong with their plan. Thus, she took full advantage of the computer learning environment: letting the computer perform a function it does well, giving feedback on the students' plans, while she focused on helping students to develop their skills in interacting with the computer.

Another example of how the teacher can help students interact with the computer learning environment comes from a study of MUMPUS, an educational game (Mandinach, 1984). This study demonstrated that the experimenter could provide hints on how to get precise feedback from the computer and illustrated an effective way for a knowledgeable adult to interact with a learner.

Another example comes from observing the behavior of expert programmers. The behavior of expert computer programmers contrasts sharply with the skills emphasized in current programming instruction as revealed by several studies of experts (Kurland & Pea, 1984; Jeffries, Turner, Pol-

son, & Atwood, 1981), and as observed by ACCCEL. To foster programming skills and higher cognitive skills, it is necessary to create a chain of cognitive accomplishments which culminates in technological expertise. Essentially, students must learn to use the skills experts use everyday. Lack of attention to this chain may, in part, account for frequent complaints of college programming teachers that instruction students received in high school actually interferes with their ability to acquire programming knowledge from post-secondary courses.

For example, precollege instruction can capitalize on template acquisition by emphasizing teaching families of template. Experts tend to learn templates by reading programs written by others. They use their templates for new problems, unless forced to invent a new one. Even then, experts attempt to reformulate similar templates rather than attempting to create whole new procedures. When a new template is created an expert might publish it so others can learn it. A study involving exactly this process is reported by Dalbey, Tourniaire, & Linn (1984).

Instruction can help students identify templates. Training that requires students to reformulate these templates is likely to build a chain of cognitive consequences that can be capitalized on by college instructors. Furthermore, instruction that requires and supports planning and testing rather than emphasizing solutions to single problems

has the potential of helping students recognize and generalize these skills.

Thus, the computer can be a flexible aid to the teacher. Setting up the conditions under which the computer aids the teacher and the teacher effectively contributes to student learning requires considerable skill. The ACCCEL project observed several teachers who have pioneered in finding effective ways to operate in computer learning environments. Computer education can be improved by providing opportunities for teachers to share their solutions to these problems through teacher networks. This approach has succeeded for writing instruction (Grey & Mayers, 1978), and has great potential for instruction involving computers.

Motivating Students

A fifth way that the computer learning environment can be capitalized on is to motivate students. The strong motivational features of the computer environment are evident in arcade parlors where students spend long hours interacting with computers. Lepper and Malone (Lepper, 1982; Malone, 1980) have identified features of the environment which make it intrinsically motivating for students. These features include three of those identified by ACCCEL: first, challenge, second, complexity, and third, interactivity. In addition, a fourth feature referred to as fantasy involvement by Lepper and Malone and characterized by the fantasy context of many computer games has been

identified. Many examples of how to capitalize on these features are available in educational games.

Recently developed educational software, such as *Rocky's Boots* (1980) from the Learning Company (see Appendix A), illustrates how motivational features can be incorporated into educational activities. *Rocky's Boots* challenges students to build machines, allows for complex solutions and provides interactive feedback. The "fantasy" of building machines may also contribute to the success of *Rocky's Boots*.

Another way that the motivating aspects of the computer can be capitalized on is in controlling access to motivating experiences. Machine access can be used as a reward for other student activities essential for maximal profit from the computer learning environment. For example, Clancy (1984), director of undergraduate computing at the University of California at Berkeley, found a way to increase the effectiveness of computer instruction at the same time as he decreased the total amount of machine access. Clancy restricted computer on-line time rather than allowing students free access to computers. He required students to come to a computer lab with a plan for the solution to their next problem as well as with a specific set of objectives for that machine interaction period. This process resulted in students using less total on-line time yet resulted in equal or greater cognitive outcomes.

POLICY IMPLICATIONS

The potential of the computer learning environment is dramatic. It is clear that the computer is a new medium of communication, not just a tool. Using this new medium to its potential requires considerable effort. The computer does not yet appear to be a savior. In fact, it may be a burden. Teachers, unsure how best to use computers and unable to get assistance may find that computers complicate the job. Eventually, by capitalizing on these complications, we can greatly increase the effectiveness of the enterprise.

To take advantage of this medium, schools need to carefully choose their policies. First, to fully appreciate the computer learning environment, schools must maintain an attitude of flexibility. The next breakthrough may require that schools reformulate their computer program. Second, to determine whether new approaches suit local needs, schools need to have an experimenting spirit. Many curricula and forms of dynamic support must be tried but only some will succeed.

Thus, effective policies for school uses of computers require better understanding of how to take full advantage of this new environment. Educators and researchers are now testing the environment and identifying effective uses of the computer. Because such tests provide much less precise feedback for educators than the computer provides for users,

this process takes considerable time. Locating examples of effective computer use and interpreting classroom experiences will build the needed expertise.

Fostering Higher Cognitive Skills

Policies which result in applying the computer learning environment to fostering higher cognitive skills offer great promise. As noted above, higher cognitive skills are infrequently emphasized in the curriculum and can be readily emphasized in computer learning environments. Taking advantage of the computer learning environment to foster higher cognitive skills addresses an important problem in education today.

The programming environment has the potential for fostering higher cognitive skills. In a subject matter area not particularly useful to students. Students learn specialized programming templates they may have little use for in the future. The skills of planning and testing are generally useful but may not transfer from the computer environment to other problems (e.g. Linn, 1983). Thus, it is particularly advantageous to find computer learning environments which foster higher cognitive skills in the subject matter areas of greatest importance to students.

Complex computer environments with interactive and precise feedback are conducive to acquisition of higher cognitive skills but difficult to create. We noted above that in

some programming environments such as LOGO, students may not see the feedback as precise because they are not able to predict what will occur on the screen. Imprecise feedback also results when software is designed to take natural language input. For example, as noted above, Aarons' (1984) software designed to foster Socratic dialogues between students and computers requires the computer to assess natural language responses to questions. This requirement severely limits the effectiveness of the software. At present, students will not receive precise feedback when using natural language for computer communication.

Thus, one way to capitalize on the computer learning environment is by fostering higher cognitive skills. Educators are encouraged to seek software in subject matter areas of importance to students. Thus far, we have only a few examples of effective subject matter specific software. Software designers are encouraged to capitalize on the experiences of others to build more powerful environments.

Fostering Basic Skills

Computers have been used extensively to foster basic skills. Often these programs have failed because the computer was not used to its potential. For example, providing drill and practice for all students in a predetermined sequence uses only the feedback function of the computer, in a manner no more powerful than popular hand-held devices such as Dataman or Speak and Spell. Unless students learn

more than just what the correct answer is they will have little idea of how to improve. In addition, drill programs are often poorly articulated with the curriculum, causing confusion rather than teaching effectively.

Thus, policies for using computers to foster basic skills need to address several issues. First, is the computer being used to full potential? Second, is the program integrated into the curriculum? A third issue concerns tradeoffs between various uses for computers. Since higher cognitive skills are infrequently exercised in most curricula, computer based programs which require these skills would seem preferable to those that do not.

Providing Dynamic Support for Learners

The computer can provide dynamic support for students thereby increasing the quality and the effectiveness of computer learning. Software selection policies which emphasize this feature will enhance the computer learning environment. As mentioned above, dynamic support is particularly useful for weaning students from reliance on the teacher and encouraging them to become more autonomous in their interactions with the computer environment.

Software designed to offer dynamic support is most effective if it provides options for students rather than insisting that students use assistance whether they want to or not. In addition, dynamic support is facilitated when

multiple modes of feedback are provided and when multiple paths to the solution of problems are available. Examples include the multiple modes available in Spider World and the multiple forms of feedback possible in programming languages.

The display feature is particularly well suited to providing dynamic support. Recent advances in feedback include dynamic displays of the flow of control as found in Rocky's Boots (1980) or in Turbo Pascal (1984). The opportunity to provide step by step information about the execution of a plan for problem solution is efficient and straightforward in the computer learning environment and may be extremely difficult in other environments. When students are asked to do desk simulations of computer programs for example, they often cannot proceed. Yet when given the dynamic displays in Turbo Pascal or the feedback from print statements inserted in their programs, they often can figure out the problem. The computer is very effective in providing the sort of dynamic support that helps students to proceed in the learning environment, but students need help to enlist the computer's aid.

In selecting software, educators can look for evidence of dynamic support. For example, the Spider World environment is essentially self contained. The initial screen provides enough information for students to proceed. In contrast, some robot worlds provide a blank screen and students

are uncertain about how to proceed. If students start out without relying on the instructor, they are already primed for autonomous learning.

Maximizing Teacher Effectiveness

Limited progress has been made in determining how the computer can flexibly aid teachers. The objective is to capitalize on the computer learning environment as well as on the skills of the teacher in order to maximize student outcomes. In general, the computer learning environment can be most effective if the teacher provides assistance in areas that the computer is not proficient. Teachers can provide clues concerning how to interact with the computer learning environment, set up the situation such that the students will engage in higher cognitive skills while interacting with the computer, and encourage students to seek assistance only if they have already tried interrogating the computer environment. More research is needed in order to better capitalize on this opportunity.

Motivating Students

The computer learning environment is particularly motivating and educators have made some progress in understanding how to capitalize on that motivation to foster student outcomes. Policies which include selection of software to take advantage of the motivating features of the environment and controlling computer access to provide reward for

autonomous behavior will be effective. Thus, progress has been made in determining how best to profit from the motivating features in the computer learning environment and policies can reflect these insights.

FUTURE DIRECTIONS

We have just begun to understand the advantages of the computer learning environment. It is possible to offer some guidelines to those designing educational programs using computers, but many questions remain unanswered. Thus far, we have a collection of ideas rather than a theory of instruction. Progress on this front is anticipated (e.g. Linn & Rohwer, 1984). While working on a theory of instruction which takes into account the characteristics of the computer learning environment, it is also appropriate to consider several additional objectives.

First, we have gained considerable understanding of the features of educational software which capitalize on the computer learning environment and much more progress is likely. Building on the success of techniques developed thus far will enable new software developers to take even greater advantage of the environment. Innovative computer uses are also needed.

Second, additional attention to the behavior of experts may well prove promising. For example, we can capitalize on the expertise gained by psychologists building expert

systems on computers. A computer simulation of an expert system provides the opportunity to match the behavior of a learner with the behavior of experts. This process is complex since some expert systems are not designed to imitate the behavior of expert humans. However, Clancy (1984) has shown how expert systems can be modified to guide the learning of novices. It may be possible, for example, to diagnose where novices go wrong and provide them guidance for improving their procedures (e.g. Langley, Ohlsson, & Sage, 1984). Thus, the computer might become a part of an educational program involving guided discovery learning. Brown (1982) offers some directions for investigating these questions.

Third, computer education decision-making needs to include consideration of the cost of computers compared to other forms of delivering instruction. Such decisions include consideration of policies which ensure equitable access to computers (e.g. Linn, 1984), and policies relating computer education experiences to vocational objectives.

Many concerns arise when schools set out to implement computer learning environments. The power of this new medium has not been adequately tapped. As the potential of the computer becomes more apparent, criticism of how schools use computers is likely to increase (e.g. Consumer's Research, 1984). Rather than placing more demands on schools, we need to provide assistance to decision makers attempting to make

this new medium effective.

Appendix A

EXAMPLES OF EFFECTIVE COMPUTER LEARNING ENVIRONMENTS

Evidence of effective use of the computer learning environment can be found in examination of successful software. Recently, software has been developed which takes advantage of the features of the computer learning environment. Primarily, new software fosters the higher cognitive skills described above in a variety of subject matter areas. The following examples are illustrative rather than exhaustive.

Algebra Arcade

A good example of how the environment can be used effectively is Algebra Arcade (Wadsworth, 1983), a modification of a program originally called Green Glabs (Dugdale, 1982) and also similar to Graphing Equations. Originally developed by Sharon Dugdale, this environment offers students the opportunity to test their ideas about algebra equations. Students can investigate questions such as: what is the role of the "a" in equations of the form " $y=ax$ " and display information on the screen to answer their question. In addition, students can engage in learning activities which require them to write equations that pass through certain points on the line. The environment encourages stu-

dents to identify problems and plan solutions to those problems by offering a practice space. Students can go to the practice space before taking their turn at the game and find out how various equations work. Students gain immediate and precise feedback for their ideas when they enter an equation.

Rocky's Boots

Rocky's Boots, a game devised by The Learning Company (1980), is another example of a software environment which capitalizes on the features of the computer. This environment provides interactive and precise feedback and fosters the higher cognitive skills of planning and testing. In Rocky's Boots, students used the primitives of "and", "or", "not", "flip-flops", and "clocks" to build logic machines which kick certain targets in a set of stimuli. To build an effective machine, students need to plan their solution. They then implement it and get precise feedback on how it works. They can modify their machines and test again, until they devise an accurate machine. Rocky's Boots is unique in its excellent provision of on-line feedback for students. Basically, once the machine is constructed, when it runs, the students can see the flow of control through the machine, can observe what each of the primitives ("and", "or", "not", "flip-flop", and "clocks") do when the flow of control reaches them, and can observe the outcome. The dynamic display features of the computer environment provide

feedback which communicates much more clearly than a static display could. This program provides a concrete representation of several abstract principles. It is possible to slow the feedback in order to find out how the machine works. As several investigations from the ACCCEL project reveal, students make progress in understanding the primitives of the game when working in this learning environment (Burbules & Reese, 1984; Linn & Stein, in press). Furthermore, direct experience with the environment is as effective as an expert tutor in fostering understanding of the primitives in the game (Linn & Stein, in press).

Computer Labs

A third software environment which provides opportunities for students to gain procedural skills and takes advantage of the computer learning environment is Atari Lab developed by Priscilla Laws. Other similar software environments include Apple Lab, the Human Relations Media Science Lab, and products developed by Dr. Robert Tinker at the Technical Education Research Center (TERC). These software tools allow students to convert the joystick or some other information gathering feature of the computer into a sensor. The sensor then is used to gather information for experiments. The programs themselves display the data and manipulate it in ways similar to those used by expert scientists. These software tools use the computer to forge a strong and immediate feedback link between the pro-

cess of collecting data and the analysis of the results. They give students the opportunity to keep the goal of the laboratory investigation in sight, to receive and respond to results in terms of that goal and to modify their experimental procedures accordingly. The computer essentially serves as an intelligent lab partner gathering and displaying data in a form which lends itself to immediate analysis, thus taking excellent advantage of the computer learning environment.

Spider world

A fourth piece of educational software which takes advantage of the features of the computer learning environment is Spider World, a program developed by John Dalbey, a member of the ACCOEL project staff (Dalbey, 1984). Spider World is a robot language similar to LOGO. Spider World differs from LOGO in that it is considerably less complex. The advantage of Spider World is that once students have mastered the small number of primitives, they can immediately be involved in planning and testing their programs. Unlike LOGO, as depicted by Kurland and Pea (1984), students using Spider World are likely to predict what will happen on the screen and be disappointed if something other than their prediction occurs. Then, for Spider World users, the environment is precise as well as interactive. In contrast, other robot worlds may not necessarily provide this preciseness for novice users. Other features of the Spider World

environment which make it advantageous for educational uses are described by Dalbey and Linn (1984).

Thus, a variety of software has been developed to take advantage of the computer learning environment. As more becomes available, software designers will be able to build on each other's successes to take even better advantage of this learning environment.

BIBLIOGRAPHY

- Anderson, J.R., Farrell, R., and Sauers, R. Learning to program in LISP. *Cognitive Science*, in press.
- Algebra Arcade. Belmont, CA: Wadsworth Electronic Publishing Company, 1983.
- Arons, A.B. Computer-based instructional dialogues in science courses. *Science*, 1984, 224, 1051-1056.
- Becker, H.J. The classroom context of microcomputers: How different schools manage the problems. Johns Hopkins University: Center for Social Organization of Schools. Paper presented at the American Educational Research Association annual meeting, New Orleans, April, 1984.
- Brown, J.S. Learning-by-doing revisited for electronic learning environments. In M.A. White (Ed.), *The Future of Electronic Learning*. Hillsdale, NJ: Lawrence Erlbaum Associates, in press.
- Burbules, N. and Reese, P. Teaching logic to children: An exploratory study of "Rocky's Boots". Berkeley, CA: ACCCEL Project, Lawrence Hall of Science, 1984.
- Carsonell, J.R. AI in CAI: An artificial intelligence approach to computer-aided instruction. *IEEE Transactions on Man-Machine Systems*, 1970, 11, 190-202.
- Chomsky, C. Finding the best language arts software. *Classroom Computer Learning*, January 1984, 4, 61-63.
- Clancy, M. Undergraduate Computer Laboratories: A Pilot Study. Berkeley, CA: University of California, 1984.
- Clancey, W. Acquiring, Representing, and Evaluating a Competence Model of Diagnostic Strategy. *The Nature of Expertise*. M. Chi, R. Glaser, M. Farr (Eds.), forthcoming.
- Clark, R.E. Learning from computers: Instructional problems. Paper presented at the American Educational Research Association annual meeting, New Orleans, April, 1984.
- Corno, L. and Mandinach, E.B. The role of cognitive engagement in classroom learning and motivation. *Educational Psychologist*, 1983, 18, 88-103.
- Dalbey, J. Spider World Reference Manual and Teacher's Guide. Berkeley, CA: ACCCEL Project, Lawrence Hall of Science, 1983.
- Dalbey, J. and Linn, M.C. The demands and requirements of computer programming: A literature review. Berkeley, CA: ACCCEL Project, Lawrence Hall of Science, 1984.

- Dalbey, J. and Linn, M.C. Spider world: A robot language for learning to program. Paper presented at American Educational Research Association annual meeting, New Orleans, 1984.
- Dalbey, J., Tourniaire, F., and Linn, M. Making Programming Instruction Cognitively Demanding: An Intervention Study. Berkeley, CA: ACCCEL Project, Lawrence Hall of Science, 1984.
- Di Sessa, A. Ezer: An integrated personal computing environment, 1984.
- Dugdale, S. Green Globbs: A Microcomputer Application for Graphing of Equations. Mathematics Teacher, March, 1982, 75.
- Federiksen, J.K., Heaver, P.A., Warren, B.M., Gillotte, H.P., Rosebery, A.S., Freeman, B., and Goodman, L. A componential approach to training reading skills. Holt Rinehart and Winston Inc., March, 1983.
- Glaser, P. Instructional Psychology: Past, Present, and Future. American Psychologist, in press.
- Gray, J. and Meyers, M. The Bay Area Writing Project. Phi Delta Kappan, 1978, pp. 410-413.
- Jeffries, R., Turner, A.D., Polson, P.G., and Atwood, M.E. The processes involved in designing software. In J.K. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1981.
- Langley, P., Ohlsson, S., and Sage, S. A machine learning approach to student modeling. Pittsburgh, Pennsylvania: The Robotics Institute, Carnegie-Mellon University, May 1, 1984.
- Lepper, M.R. Microcomputers in Education: Motivational and Social Issues. Presented to the 90th annual meeting of the American Psychological Association, Washington, D.C., 1982.
- Levin, H. Newsletter. Stanford, CA: Institute for Finance & Governance, 1984.
- Levin, J.A., Riel, M.M., Rowe, R.D., and Boruta, M.J. Muktuk meets jacuzzi: Computer networks and elementary school writers. In S.W. Freedman (Ed.), The acquisition of written language: Evidential and response. Hillsdale, N.J.: Ablex, in press.
- Linn, M.C. Fostering equitable consequences from computer learning environments. Sex Roles, in press.
- Linn, M.C. Redesigning science education: What is the role of science education research?. In R. Bybee (Ed.), Redesigning science and technology education. Washington D.C.: National

Science Teachers Association Yearbook, 1984.

- Linn, M.C. Science curriculum design: Views from a psychological framework. *Psychology and curriculum design*. R. Dillon and R. Sternberg (Eds.) New York: Academic Press, in press.
- Linn, M.C. and Fisher, C.W. The gap between promise and reality in computer education: Planning a response. *Making our schools more effective*. San Francisco, CA: Far West Laboratories, 1984.
- Linn, M.C., Fisher, C.W., Dalbey, J., Mandinach, E.B., Burbules, M.C., and Reese, P. *The effects of learning to program and learning from cognitively demanding software*. Berkeley, CA: ACCCEL Project, Lawrence Hall of Science, 1983.
- Linn, M.C., Fisher, C., Mandinach, E.B., Dalbey, J., and Beckum, L. *Demands and cognitive consequences of computer learning*. NIE Proposal OE 0460830017, 1982.
- Linn, M.C. and Rohwer, W. *Effects of autonomous learning on cognitive skill acquisition in computer programming courses*. Proposal to the National Science Foundation, 1984.
- Linn, M.C. and Stein, J.S. *Promise and reality in computer education*. *Instructional Innovator*, in press.
- Malone, T.W. *What makes things fun to learn? A study of intrinsically motivating computer games*. Palo Alto, Ca: XEROX Palo Alto Research Center, August, 1980.
- Mandinach, E.B. *The role of strategic knowledge and self-regulation in learning to solve an intellectual computer game*. Unpublished doctoral dissertation, Stanford University, 1984.
- McKeithen, K.B., Reitman, J., Rueter, H., and Hirtle, S. Knowledge organization and skill differences in computer programmers. *Cognitive Psychology*, 1981, 13, 307-325.
- National Commission on Excellence in Education. *A nation at risk: The imperative for education reform*. Washington, D.C.: U.S. Government Printing Office, 1983.
- National Science Board Commission on Pre-college Education in Mathematics, Science, and Technology. *Educating Americans for the 21st Century*, September, 1983.
- Nickerson, R.S. *Computer programming as a vehicle for teaching thinking skills. Three Uses of Computers in Education*. Washington D.C.: National Institute of Education, 1984.
- Norman, D.A. and Rumelhart, D.E. *Explorations in cognition*. San Francisco: W.H. Freeman & Co., 1975.

- Papert, S. Mindstorms. New York: Basic Books, Inc. , 1981.
- Pea, R.D. and Kurland, D.M. On the cognitive prerequisites of learning computer programming. New York : Bank Street College of Education, April 15, 1983.
- Riel, M. Education and ecstasy: Computer chronicles of students writing together. The quarterly newsletter of the laboratory of comparative human cognition, 2, 59-67. 1983.
- Riel, M.M., Levin, J.A., and Miller-Souviney, B. Dynamic support and educational software development. Paper presented at the American Educational Research Association annual meeting, New Orleans, April, 1984.
- Rocky's Boots. Menlo Park, CA: The Learning Company, 1980.
- The case against the classroom computer. Consumers' research magazine, March, 1984.
- Salomon, G. and Gardner, H. The Computer as Educator: Lessons from television Research. Paper presented at the American Educational Research Association annual meeting, New Orleans, April, 1984.
- Sleeman, D. and Brown, J.S. Intelligent Tutoring Systems. New York: Academic Press, 1982.
- Taylor, R. The Computer in the School: Tutor, Tool, Tutor. New York: Teachers College Press, 1980.
- Thomas, J.W. and Rohwer, W.D. The relationship between study activity and achievement as a function of instructional requirements and task latitude. San Francisco: Far West Laboratory for Educational Research and Development, Research Monograph No. 50, 1982.
- Turbo Pascal. Creative Computing, July, 1984, 12, 53-54.